Proceedings of the 22nd UK Conference of the
Association for Computational Mechanics in Engineering
2 - 4 April 2014, University of Exeter, Exeter

A REVIEW OF THE MATERIAL POINT METHOD AND ITS LINKS TO OTHER COMPUTATIONAL METHODS

***T.J. Charlton, W.M. Coombs and C.E. Augarde**

Mechanics Research Group, School of Engineering & Computing Sciences, Durham University, South Road, Durham , DH1 3LE, UK

*t.j.charlton@durham.ac.uk

ABSTRACT

There is considerable interest in development of solid mechanics modelling which can cope with both material and geometric nonlinearity, particularly in areas such as computational geotechnics, for applications such as slope failure and foundation installation. One such technique is the Material Point Method (MPM), which appears to provide an efficient way to model these problems. The MPM models a problem domain using particles at which state variables are kept and tracked. The particles have no restriction on movement, unlike in the Finite Element Method (FEM) where element distortion limits the level of mesh deformation. In the MPM, calculations are carried out on a regular background grid to which state variables are mapped from the particles. It is clear, however, that the MPM is actually closely related to existing techniques, such as ALE and in this paper we review the MPM for solid mechanics and demonstrate these links.

Key Words: *Material Point Method; Solid Mechanics; Mesh free; Finite Deformation Mechanics*

1. Introduction

Most computational methods in solid mechanics can be described as either Eulerian or Lagrangian. Lagrangian methods work by splitting a problem into elements or particles in a mesh, and throughout any deformation this mesh follows the problem domain. The advantage of this is that it is easy to track surfaces and history dependant variables throughout a simulation as the position in relation to other elements is always maintained. A disadvantage of a Lagrangian approach is that problems can begin to occur when a material undergoes large deformations. These large deformations can result in a heavily distorted mesh which, in certain situations, can result in calculations being unable to be completed. Eulerian methods work by having a mesh that is fixed in space and allowing particles to move within it. This is more commonly used in fluid mechanics applications but overcomes any issue of mesh distortion as the mesh remains the same throughout calculations. A disadvantage is that it becomes more difficult to track boundaries and history dependent variables as particles move. Attempts have been made to combine together Eulerian and Lagrangian methods with the aim of keeping the positives without the drawbacks. One method to combine these features is the Material Point Method (MPM) [1]. There are currently many uses of the MPM, often in situations where the FEM struggles due to highly distorted elements. Some uses currently of interest include problems involving impact and collision, penetration, crack propagation, slope stability, soil mechanics and simulation of snow for use in animation. In this paper we will review the MPM and highlight links to other techniques such as the Finite Element (FE) and Arbitrary Lagrangian Eulerian (ALE) methods.

2. Method Overview

The Material Point Method (MPM) was first developed by Sulsky et al. [1] as an extension for solid mechanics of the FLuid Implicit Particle (FLIP) [2] method, which itself was an extension to the Particle in Cell (PIC) [3] method used in fluid dynamics. The MPM can be referred to as a meshfree method.

Although a background grid of connected nodes is required to perform calculations, material properties are carried by a series of particles which are free to move independently of each other. In the MPM, material points, known as particles, store state variables and move through a background grid or mesh which can be changed or reset following each time step or load increment. This can be seen in Figure 1 where a material has been deformed and the mesh has been reset with particles in updated positions.

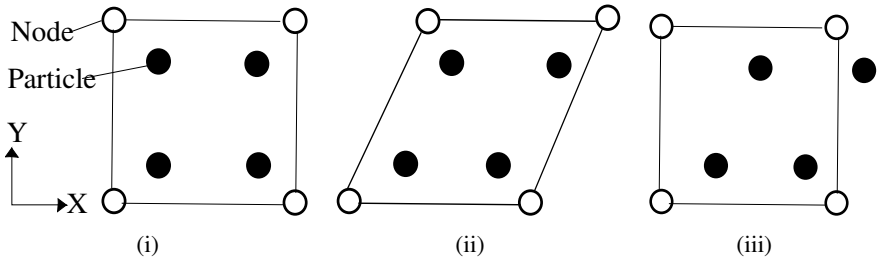

Figure 1: Particle positions as mesh deforms and is reset: (i) original configuration, (ii) deformed mesh and (iii) reset mesh

Initially a material domain is split into a number of elements similar to the FEM. Each of these elements is then populated with a number of material points. Each material point is assigned a weight based on the volume of material that the particle represents. It can be desirable to initially locate particles at Gauss quadrature points to simplify the volume weight calculation. In addition to the mesh covering the material's initial position, the mesh must extend to where the material is expected to deform.

In each element containing particles, the state variables must be mapped from the particles to the grid nodes. This mapping process is carried out within each element using shape functions similar to those used in the FEM. For instance the external force at a grid node is given by

$$f_g^{ext} = \sum_{i=1}^{n_{mp}} N_i f_{p_i}^{ext} \tag{1}$$

where f_p^{ext} is the particle external force, N_i are the nodal shape functions for the element containing the particle and n_{mp} refers to the number of material points in the grid element.

To be able to map to the correct grid nodes it is necessary to know in which element each material point located at a point in time. Although trivial initially, after particles have moved this problem can become more complex, especially if the mesh is not uniform. To simplify this process it is common to reset the background mesh to a uniform grid after each loadstep.

The stiffness of each element is determined from the contributions from each of the particles currently inside. Once the global stiffness matrix is assembled and the grid node displacements determined, the grid node displacements $\{u_g\}$ are then mapped back to particles to get particle displacements $\{u_p\}$ through

$$\{u_p\} = \sum_{i=1}^{n_n} N_i \{u_{g_i}\}, \tag{2}$$

where n_n is the number of element nodes. The particle positions are then updated. The grid node displacements are not used to update the position of nodes in the mesh; the original undeformed mesh is used.

3. Comparison with FEM

The MPM has many similarities with the standard FEM, in fact it is possible to think of the MPM as the same as the FEM but with moving integration points instead of fixed Gauss points in each element. The shape functions used to map between the grid and the particles in the MPM are the same as the shape functions used in the FEM. If material points are located at the positions of Gauss points in the FEM and if the mesh is not reset after each step then the MPM becomes identical to the FEM. In the MPM, external forces can be applied directly to the grid nodes or can be applied at particles and mapped to the grid, however to do this particles must be placed where forces are applied.

Boundary conditions are applied directly to the background mesh as in the FEM. This works for fixed boundaries with zero displacement conditions however to track boundaries it is necessary for some particle positions to coincide with material boundaries. Currently it is difficult to do this.

As the particles move it is possible that the background element they are in changes. One of the main problems with the MPM is an error which occurs when material points cross element boundaries causing an imbalance of internal forces [4]. In the MPM, if an element becomes void of particles it is turned off so that it no longer contributes to the global stiffness matrix.

Due to the background nodal relationships the MPM is not as computationally expensive as some meshless methods. However, as there is an additional mapping step, it is more expensive than the standard FEM. In regular finite elements the Jacobian has to be calculated only once however in MPM as the particles move and are not in consistent positions it is necessary to calculate the Jacobian for each material point.

4. Comparison with Arbitrary Lagrangian-Eulerian Methods

Arbitrary Lagrangian Eulerian (ALE) Methods [5], like the MPM, take advantage of useful aspects of both types of method while trying to avoid disadvantages. In ALE methods a mesh is allowed to move independently of the material moving in an arbitrary manner that can be defined by the user. This is achieved by having a third set of reference coordinates other than the initial and current configurations, this allows the mesh to be adapted to avoid problems caused by mesh distortions in a purely Lagrangian method. In the MPM the grid can be adapted in a similar manner if desired however it is usual to reset the grid to an undeformed uniform state as this removes the additional expense of particle searching.

5. Comparison with Smoothed Particle Hydrodynamics

Smoothed particle hydrodynamics (SPH) [6] is a Lagrangian meshfree method, where the domain is represented by a set of distributed particles needing no connectivity, rather than split up into a set of nodes in a grid. Like the MPM, each particle possesses a set of material properties and moves according to governing equations. However SPH doesn't have a background grid where calculations take place. Instead, field functions at these points are approximated using a kernel function. These approximations are then smoothed using a weighted average over neighbouring particles. Having no background grid, shape functions cannot be used in the same way as in FEM and MPM. To calculate the support and influence domains of a particle a weighting function is used. This is common to most meshfree methods.

A particular advantage of SPH is its ability to handle large deformations. This is due to the fact that particles aren't restrained to a mesh. However it is not as straightforward to apply boundary conditions when using SPH for solid mechanics. The need to search for a nearest neighbour to define nodal connectivity can also make SPH more computationally expensive than the standard MPM.

6. Numerical Example

Due to the similarities between the FEM and the MPM, a compact finite element code has been used as the basis for a MPM code [7]. Material points, which originally were located at Gauss points are freed to move after each load step and then the mesh is reset to its original configuration. Because of these similarities between methods, the constitutive model used in the FEM can also be used in the MPM. A Total Lagrangian FEM code has been modified to facilitate movement of material points, however problems have been noted when material points cross grid element boundaries. As mentioned above, this is a common issue highlighted in the literature.

The code was used for the simple problem of one-dimensional compression of a $1\times1\times1m$ cube, as shown in Figure 2(i). A Young's modulus $E = 1\times10^9 Pa$ and Poissons ratio of $\nu = 0.2$ were used with downward forces totalling $5\times10^9 N$ applied to the four uppermost particles over 10 loadsteps. The particle positions within two elements in the Z direction throughout the simulation are shown in Figure 2(ii). It can be seen that a problem arises when the uppermost particles displace into the lower element. When this happens the internal force calculations result in a displacement back into the previous element for some of the particles. This is repeated over the following loadsteps resulting in a oscillation of particles between two elements.

One method that addresses the problem that occurs when particles cross element boundaries is the GIMP method where characteristic functions are assigned to particles which give particles a support area so that

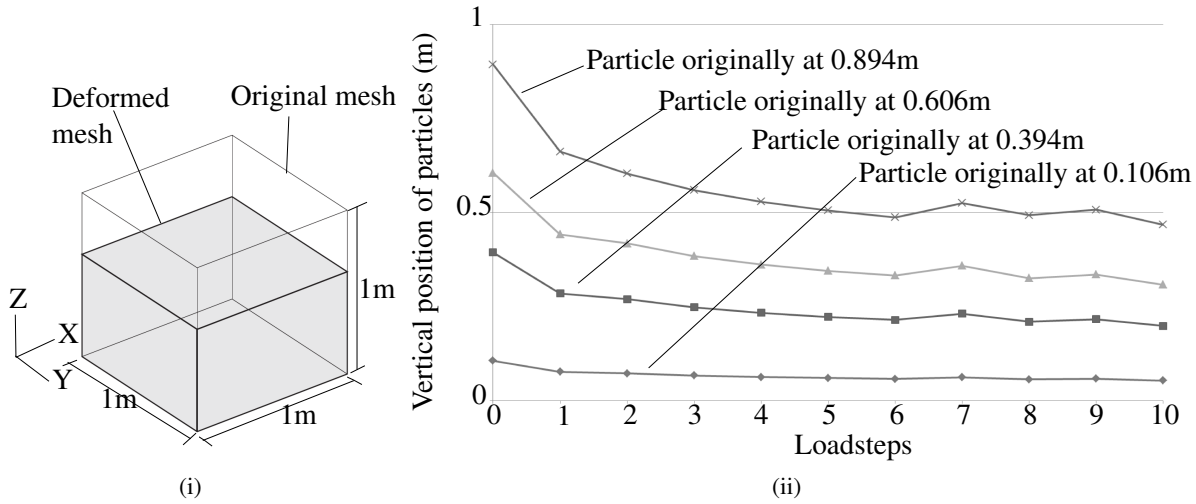

Figure 2: One dimensional compression of a unit cube: (i) Original and deformed mesh. (ii) Position of particles within 2 elements over the period of 10 loadsteps.

a particle can affect nodes in cells other than the one where the particle is located. The computational expense of GIMP is larger than that of standard MPM however it provides improved accuracy stability and robustness to simulations [8]. This is the next step in this work.

7. Conclusions and future work

From the work undertaken so far, the MPM appears to a promising technique for dealing with large deformations, that can be simply achieved by altering an existing FE code. A MPM code is being developed using an Updated Lagrangian approach so that calculations are not affected by a stiffness that was calculated when material points were in different positions. It is hoped that in the future a GIMP type method can be implemented to address the boundary crossing problem encountered, and to then use the MPM code for more complex problems involving large deformations.

References

[1] D. Sulsky, Z. Chen, and H. Schreyer, "A particle method for history-dependent materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 118, pp. 179 – 196, 1994.

[2] J. Brackbill and H. Ruppel, "Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions," *Journal of Computational Physics*, vol. 65, no. 2, pp. 314 – 343, 1986.

[3] M. W. Evans, F. H. Harlow, and E. Bromberg, "The particle-in-cell method for hydrodynamic calculations," DTIC Document, Tech. Rep., 1957.

[4] M. Steffen, R. M. Kirby, and M. Berzins, "Analysis and reduction of quadrature errors in the material point method (mpm)," *International Journal for Numerical Methods in Engineering*, vol. 76, no. 6, pp. 922–948, 2008.

[5] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodrguez-Ferran, *Arbitrary Lagrangian Eulerian Methods*. John Wiley & Sons, Ltd, 2004.

[6] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *The Astronomical Journal*, vol. 82, pp. 1013–1024, 1977.

[7] W. Coombs, R. Crouch, and C. Augarde, "70-line 3d finite deformation elastoplastic finite-element code," in *Numerical Methods in Geotechnical Engineering*. CRC Press, Jun. 2010, pp. 151–156.

[8] S. Bardenhagen and E. Kober, "The generalized interpolation material point method," *Computer Modeling in Engineering and Sciences*, vol. 5, no. 6, pp. 477–496, 2004.